Scale-Adaptive Group Optimization for Social Activity Planning

Hong-Han Shuai[1(✉)], De-Nian Yang[2], Philip S. Yu[3], and Ming-Syan Chen[1,2]

[1] Graduate Institute of Communication Engineering,
National Taiwan University, Taipei, Taiwan
{hhshuai,dnyang}@iis.sinica.edu.tw
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan
mschen@cc.ee.ntu.edu.tw
[3] Department of Computer Science, University of Illinois at Chicago, Chicago, USA
psyu@uic.edu

Abstract. Studies have shown that each person is more inclined to enjoy a group activity when 1) she is interested in the activity, and 2) many friends with the same interest join it as well. Nevertheless, even with the interest and social tightness information available in online social networks, nowadays many social group activities still need to be coordinated manually. In this paper, therefore, we first formulate a new problem, named Participant Selection for Group Activity (PSGA), to decide the group size and select proper participants so that the sum of personal interests and social tightness of the participants in the group is maximized, while the activity cost is also carefully examined. To solve the problem, we design a new randomized algorithm, named Budget-Aware Randomized Group Selection (BARGS), to optimally allocate the computation budgets for effective selection of the group size and participants, and we prove that BARGS can acquire the solution with a guaranteed performance bound. The proposed algorithm was implemented in Facebook, and experimental results demonstrate that social groups generated by the proposed algorithm significantly outperform the baseline solutions.

1 Introduction

Studies have shown that two important factors are usually involved in a person's decision to join a social group activity: (1) interest in the activity topic or content, and (2) social tightness with other attendees [5,8]. For example, if a person who appreciates jazz music has complimentary tickets for a jazz concert in Rose Theatre, she is inclined to invite her friends or friends of friends who are also jazzists. However, even the information on the two factors is now available online, the attendees of most group activities still need to be selected manually, and the process will be tedious and time-consuming, especially for a large social activity, given the complicated social link structure and the diverse interests of potential attendees.

T. Cao et al. (Eds.): PAKDD 2015, Part I, LNAI 9077, pp. 45–57, 2015.
DOI: 10.1007/978-3-319-18038-0_4

Recent studies have explored community detection, graph clustering and graph partitioning to identify groups of nodes mostly based on the graph structure [1]. The quality of an obtained community is usually measured according to its internal structure, together with its external connectivity to the rest of the nodes in the graph [7] . Those approaches are not designed for activity planning because it does not consider the interests of individual users along with the cost of holding an activity with different numbers of participants. An event which attracts too few or too many attendees will result in unacceptable loss for the planner. Therefore, it is important to incorporate the preference of each potential participant, their social connectivity, and the activity cost during the planning of an activity.

With this objective in mind, a new optimization problem is formulated, named *Participant Selection for Group Activity (PSGA)*. The problem is given a cost function related to the group size and a social graph G, where each node represents a potential attendee and is associated with an interest score that describes the individual level of interest. Each edge has a social tightness score corresponding to the mutual familiarity between the two persons. Since each participant is more inclined to enjoy the activity when 1) she is interested in the activity, and 2) many friends with the same interest join as well, the *preference* of a node v_i for the activity can be represented by the sum of its interest score and social tightness scores of the edges connecting to other participants, while the *group preference* is sum of the total interest scores of all participants and the social tightness scores of the edges connecting to any two participants. Moreover, the *group utility* here is represented by the group preference subtracted by the *activity cost* (ex. the expense in food and siting), which is usually correlated to the number of participants.[1] The objective of PSGA is to determine the best group size and select proper participants, so that the group utility is maximized. In addition, the induced graph of the set F of selected participants is desired to be a connected component, so that each attendee is possible to become acquainted with another attendee according to a social path[2].

One possible approach to solving PSGA is to examine every possible combination on every group size. However, this enumeration approach of group size k requires the evaluation of C_k^n candidate groups, where n is the number of nodes in G. Therefore, the number of group size and attendee combinations is $O(2^n)$, and it thereby is not feasible in practical cases. Another approach is to incrementally construct the group using a greedy algorithm that iteratively tries each group size and sequentially chooses an attendee that leads to the largest increment in group utility at each iteration. However, greedy algorithms are inclined to be trapped in local optimal solutions. To avoid being trapped in local optimal

[1] Different weighted coefficients can be assigned to the group utility and activity cost according to the corresponding scenario.

[2] For some group activities, it is not necessary to ensure that F leads to a connected subgraph, and those scenarios can be handled by adding a virtual node v connecting to every other node in G, and choosing v in F for PSGA always creates a connected subgraph in $G \cup \{v\}$, but F may not be a connected subgraph in G.

solutions, randomized algorithms have been proposed as a simple but effective strategy to solve problems with large instances [12].

A simple randomized algorithm is to randomly choose multiple start nodes initially. Each start node is considered as a partial solution, and a node neighboring the partial solution is randomly chosen and added to the partial solution at each iteration later. Nevertheless, this simple strategy has three disadvantages. Firstly, a start node that has the potential to generate final solutions with high group utility does not receive sufficient computational resources for randomization in the following iterations. More specifically, each start node in the randomized algorithm is expanded to only one final solution. Thus, a good start node will usually fail to generate a solution with high group utility since it only has one chance to randomly generate a final solution. The second disadvantage is that the expansion of the partial solution does not differentiate the selection of the neighboring nodes. Each neighboring node is treated equally and chosen uniformly at random for each iteration. Even this issue can be partially resolved by assigning the selection probability to each neighboring node according to its interest score and the social tightness of incident edges, this assignment will lead to the greedy selection of neighbors and thus tends to be trapped in local optimal solutions as well. The third disadvantage is that the linear scanning of different group sizes is not computationally tractable for real scenarios as an online social network contains an enormous number of nodes.

Keeping the above observations in mind, we propose a randomized algorithm, called *Budget-Aware Randomized Group Selection (BARGS)*, to effectively select the start nodes, expand the partial solutions, and estimate the suitable group size. The computational budget represents the target number of random solutions. Specifically, *BARGS* first selects a group size limit k_{max} in accordance with the cost function[3]. Afterward, m start nodes are selected, and neighboring nodes are properly added to expand the partial solution iteratively, until k_{max} nodes are included, while the group size corresponding to the largest group utility is acquired finally. Each start node in *BARGS* is expanded to multiple final solutions according to the assigned budget. To properly invest the computational budgets, each stage of *BARGS* invests more budgets on the start nodes and group sizes that are more inclined to generate good final solutions, according to the sampled results from the previous stages. Moreover, the node selection probability is adaptively assigned in each stage by exploiting the cross entropy method. In this paper, we show that our allocation of computation budgets is the optimal strategy, and prove that the solution acquired by *BARGS* has a guaranteed performance bound.

The rest of this paper is organized as follows. Section 2 formulates PSGA and surveys related works. Sections 3 explains *BARGS* and derives the performance bound. User study and experimental results are presented in Section 4, and we conclude this paper in Section 5.

[3] For instance, if the largest capacity of available stadiums for a football game is $20,000$, k_{max} is set as $20,000$.

2 Preliminary

2.1 Problem Definition

Given a social network $G = (V, E)$, where each vertex $v_i \in V$ and each edge $e_{i,j} \in E$ are associated with an interest score η_i and a social tightness score $\tau_{i,j}$ respectively, we study a new optimization problem for finding a set F of vertices which maximizes the *group utility* $U(F)$, i.e.,

$$U(F) = \sum_{v_i \in F} (\eta_i + \sum_{v_j \in F : e_{i,j} \in E} \pi_{i,j}) - \beta C(|F|), \tag{1}$$

where F with $|F| \leq k_{max}$ is a connected subgraph in G to encourage each attendee to be acquainted with another attendee with at least one social path in F, C is a non-negative activity cost function based on the number of attendees, and β is a weighted coefficient between the preference and cost. For each node v_i, let $\eta_i + \sum_{v_j \in F : e_{i,j} \in E} \pi_{i,j}$ denote the *preference* of node v_i on the social group activity[4]. PSGA is very challenging due to the tradeoff between interest, social tightness, and the cost function, while the constraint assuring that F is connected also complicates this problem because it is no longer able to arbitrarily choose any nodes from G. Indeed, we show that PSGA is NP-hard in [15].

2.2 Related Works

A recent line of study has been proposed to find cohesive subgroups in social networks with different criteria, such as cliques, n-clubs, k-core, and k-plex. Saríyüce et al. [14] proposed an efficient parallel algorithm to find a k-core subgraph, where every vertex is connected to at least k vertices in the subgraph. Xiang et al. [16] proposed a branch-and-bound algorithm to acquire all maximal cliques that cannot be pruned during the search tree optimization. Moreover, finding the maximum k-plexes was comprehensively discussed in [11]. On the other hand, community detection and graph clustering have been exploited to identify the subgraphs with the desired structures [1]. The quality of a community is measured according to the structure inside the community and the structure between the community and the rest of the nodes in the graph, such as the density of local edges, deviance from a random null model, and conductance [7]. Nevertheless, the above models did not examine the interest score of each user and the social tightness scores between users, which have been regarded as crucial factors for social group activities. Moreover, the activity cost for the group is not incorporated during the evaluation.

In addition to dense subgraphs, social groups with different characteristics have been explored for varied practical applications. Expert team formation in social

[4] Different weights λ and (1-λ) can be assigned to the interest scores and social tightness such that $U(F) = \sum_{v_i \in F} (\lambda_i \eta_i + (1 - \lambda_i) \sum_{v_j \in F : e_{i,j} \in E} \tau_{i,j}) - \beta C(|F|)$. λ_i can be set directly by a user or according to the existing model [18]. The impacts of different λ will be studied later in Section 4.

networks has attracted extensive research interest. The problem of constructing an expert team is to find a set of people possessing the required skills, while the communication cost among the chosen friends is minimized to optimize the rapport among the team members to ensure efficient operation. Communication costs can be represented by the graph diameter, the size of the minimum spanning tree, and the total length of the shortest paths [9]. Finding influential event organizers who can influence largest number of attendees to join the event is studied [6]. By contrast, minimizing the total spatial distance with R-Tree from the group with a given number of nodes to the rally point is also studied [17]. Nevertheless, this paper focuses on a different scenario that aims at identifying a group with the most suitable size according to the activity cost, while those selected participants also share the common interest and high social tightness.

3 Algorithm Design for PSGA

To solve PSGA, a baseline approach is to incrementally constructing the solution by sequentially choosing and adding a neighbor node that leads to the largest increment in the group preference until k_{max} people are selected. Afterward, we derive the group utility for each k by incorporating the activity cost, $1 \leq k \leq k_{max}$, and extract the group size k^* with the maximum group utility. The theoretical analysis of greedy algorithm is presented in [15] due to the space constraint.

The greedy algorithm, despite the simplicity, the search space of the greedy algorithm is limited and thus tends to be trapped in a local optimal solution, because only a single sequence of solutions is explored. To address the above issues, this paper proposes a randomized algorithm *BARGS* to randomly choose m start nodes[5]. *BARGS* leverages the notion of Optimal Computing Budget Allocation (OCBA) [3] to systematically generate the solutions from each start node, where the start nodes with more potential to generate the final solutions with large group utility will be allocated with more budgets (i.e., expanded to more final solutions). In addition, since each start nodes can generate the final solutions with different group sizes, the size with larger group utility will be associated with more budgets as well (i.e., generated more times). Specifically, *BARGS* includes the following two phases.

1) Selection and Evaluation of Start Nodes and Group Sizes: This phase first selects m start nodes according to the summation of the interest scores and social tightness scores of incident edges. Each start node acts as a seed to be expanded to a few final solutions. At each iteration, a partial solution, which consists of only a start node at the first iteration or a connected set of nodes at each iteration afterward, is expanded by randomly selecting a node neighboring to the partial solution, until $k_{\max}$ nodes are included. The group utility of each intermediate and final solution is evaluated to optimally allocate different computational budgets to different start nodes and different group sizes in the next phase.

[5] The impact of m will be studied in Section 4.

2) Allocation of Computational Budgets: This phase is divided into r stages[6], while each stage shares the same total computational budget. In the first stage, the computational budget allocated to each start node is determined by the sampled group utility in the first phase. In each stage afterward, the computational budget allocated to each start node is adjusted by the sampled results in the previous stages. Note that each node can generate different numbers of final solutions with different group sizes. The sizes with small group utility sampled in the previous stages will be associated with smaller computational budgets in the current stage. Therefore, if the activity cost is a convex cost function, the cost increases more significantly as the group size grows, and *BARGS* tends to allocate smaller computational budgets and thus generates fewer final solutions with large group sizes.

During the expansion of the partial solutions, we differentiate the probability to select each node neighboring to a partial solution. One intuitive way is to associate each neighboring node with a different probability according to the sum of the interest scores and social tightness scores on the incident edges. Nevertheless, this assignment is similar to the greedy algorithm as it limits the scope to only the local information, making it difficult to generate a final solution with large group utility. By contrast, *BARGS* exploits the cross entropy method [13] according to sampled results in the previous stages in order to optimally assign a probability to the edge incident to a neighboring node.

Due to the space constraint, the detailed pseudocode is presented in [15]. In the following, we first present how to optimally allocate the computational budgets to different start nodes and different group sizes. Afterward, we exploit the cross entropy method to differentiate the neighbor selection during the expansion of the partial solutions. The performance bound and illustrative example of the proposed algorithm are provided in the full version [15].

Allocation of Computational Budgets. Similar to the baseline greedy algorithm, allocating more computational budgets to a start node v_i with larger group preference (i.e., $\sum_{v_i \in F}(\eta_i + \sum_{v_j \in F: e_{i,j} \in E} \pi_{i,j})$) examines only the local information and thus is difficult to generate the solution with large group utility. Therefore, to optimally allocate the computational budgets for each start node and each group size, we first define the solution quality as follows.

Definition 1. *The solution quality, denoted by Q, is defined as the maximum group utility of the solution generated from the m start nodes among all sizes.*

For each stage t of phase 2 in *BARGS*, let $N_{i,k,t}$ denote the computational budgets allocated to the start node v_i with size k in the t-th stage. In the following, we first derive the optimal ratio of the computational budgets allocated to any two start nodes v_i and v_j with group size k and l, respectively. Let two random variables $Q_{i,k}$ and $Q^*_{i,k}$ denote the sampled group utility of any solution

[6] The detailed settings of the parameters of the algorithm, such as m, r, α, and β are presented in [15].

and the maximal sampled group utility of a solution for start node v_i with size k, respectively. If the activity cost is not considered, according to the central limit theorem, $Q_{i,k}$ follows the normal distribution when $N_{i.k}$ is large, and it can be approximated by the uniform distribution in $[c_{i,k}, d_{i,k}]$ as analyzed in OCBA [3], where $c_{i,k}$ and $d_{i,k}$ denote the minimum and maximum sampled group utility in the previous stages, respectively. On the other hand, when the activity cost is considered, the cumulative distribution function is shifted by $C(k)$, and it still follows the same distribution. Therefore, we have the following lemma.

Lemma 1. *Assume that $d_{j,l} \geq c_{i,k}$, the probability that the solution generated from the start node v_i with size k is better than the solution generated from the start node v_j with size l, i.e., $P(Q^*_{i,k} \leq Q^*_{j,l})$, is at least $\frac{1}{2}(\frac{d_{j,l}-c_{i,k}}{d_{i,k}-c_{i,k}})^{N_{i,k}}$.*

Proof. Due to the space constraint, the detailed proof is presented in [15].

Let v_b and k^*_b denote the best start node and best activity size for v_b, respectively. The ratio between $N_{i,k,t}$ and $N_{j,l,t}$ equals $P(Q^*_{i,k} \geq Q^*_{b,k^*_b}) : P(Q^*_{j,l} \geq Q^*_{b,k^*_b})$, which is optimal as shown in OCBA [3]. However, the computational costs for different group sizes are not the same, e.g., the computational cost of the total group utility for size 1 is much smaller than the computational cost for size 100. Since the computational complexity of adding a node to a partial solution of size $k-1$ is $O(k)$, we derive the ratio of the computational budgets between $N_{i,k,t}$ and $N_{j,l,t}$ as follows.

$$\frac{N_{i,k,t}}{N_{j,l,t}} = \frac{\frac{1}{k} \cdot P(Q^*_{i,k} \geq Q^*_{b,k^*_b})}{\frac{1}{l} \cdot P(Q^*_{j,l} \geq Q^*_{b,k^*_b})}. \tag{2}$$

Note that if the allocated computational budgets for a start node is 0 in the t-th stage, we prune off the start node in the any stage afterward. Moreover, when we generate a solution with group size k, the solutions from size 1 to size $k-1$ are also generated as well. Therefore, to avoid generating an excess number the solutions with small group sizes, it is necessary to relocate the computation budgets. Let $\hat{N}_{i,k,t}$ denote the reallocated budget of start node v_i with size k in the t-th stage. *BARGS* derives $N_{i,k,t}$ as follows.

$$\hat{N}_{i,k,t} = \max(0, N_{i,k,t} - \sum_{l>k} \hat{N}_{i,l,t}). \tag{3}$$

Specifically, after deriving $N_{i,k,t}$ with Eq. 2, *BARGS* derives $\hat{N}_{i,k,t}$ from $k = k_{max}$ to 1. Initially, $\hat{N}_{i,k_{\max},t} = N_{i,k_{\max},t}$. Afterward, for $k = k_{max} - 1$, if $N_{i,k_{max}-1,t}$ is equal to $N_{i,k_{max},t}$, it is not necessary to generate additional solutions with size $k_{max}-1$ since they have been created during the generation of the solutions with size k_{max}. In this case, $\hat{N}_{i,k_{max}-1,t}$ is 0. Otherwise, *BARGS* sets $\hat{N}_{i,k_{max}-1,t} = N_{i,k_{max}-1,t} - \hat{N}_{i,,k_{max},t}$. The above process repeats until $k = 1$. Since the number of solutions with size k is still $N_{i,k,t}$, the computational budget allocation is still optimal as shown in Eq. 2.

Neighboring Node Differentiation. To effectively differentiate neighbor selection, *BARGS* exploits the cross entropy method [13] to achieve importance sampling by adaptively assigning a different probability to each neighboring node from the sampled results in previous stages.

Definition 2. *A Bernoulli sample vector, denoted as* $X_{i,k,q} = \langle x_{i,k,q,1}, ..., x_{i,k,q,j}, ..., x_{i,k,q,n} \rangle$*, is defined to be the q-th sample vector from start node* v_i*, where* $x_{i,k,q,j}$ *is* 1 *if node* v_j *is selected in the q-th sample and* 0 *otherwise.*

Take start node v_i with size k as an example, after collecting $N_{i,k,1}$ samples $X_{i,k,1}, X_{i,k,2}, ..., X_{i,k,q}, ..., X_{i,k,N_{i,k,1}}$ generated from start node v_i, *BARGS* calculates the total group utility $U(X_{i,k,q})$ for each sample and sorts them in the descending order, $U_{(1)} \geq ... \geq U_{(N_{i,k,1})}$. Let $\gamma_{i,k,1}$ denotes the group utility of the top-ρ performance sample, i.e. $\gamma_{i,k,1} = U_{(\lceil \rho N_{i,k,1} \rceil)}$. With those sampled results, we set the selection probability $p_{i,k,t+1,j}$ of every node v_j in iteration $t+1$ for the partial solution expanded from node v_i by fitting the distribution of top-ρ performance samples as follows.

$$p_{i,k,t+1,j} = \frac{\sum_{q=1}^{N_{i,k,t}} I_{\{U(X_{i,k,q}) \geq \gamma_{i,k,t}\}} x_{i,k,q,j}}{\sum_{q=1}^{N_{i,k,t}} I_{\{U(X_{i,k,q}) \geq \gamma_{i,k,t}\}}}, \tag{4}$$

where $I_{\{U(X_{i,k,q}) \geq \gamma_{i,k,t}\}}$ is 1 if the group utility of sample $X_{i,k,q}$ is no smaller than a threshold $\gamma_{i,k,t} \in \mathbb{R}$, and 0 otherwise. Intuitively, the neighbor that tends to generate a better solution will be assigned a higher selection probability. As shown in [13], the above probability assignment scheme has been proved to be optimal from the perspective of cross entropy. Eq. 4 minimizes the Kullback-Leibler cross entropy (KL) distance between node selection probability and the distribution of top-ρ performance samples, such that the performance of random samples in the $(t+1)$-th stage is guaranteed to be closest to the top- ρ performance samples in the t-th stage. Due to the space constraint, the illustrative example and theoretical results are provided in the full version [15].

Time Complexity of BARGS. The time complexity of *BARGS* contains two parts. The first phase selects m start nodes with $O(E+n+ m \log n)$ time, where $O(E)$ is to sum up the interest and social tightness scores, $O(n + m \log n)$ is to build a heap and extract m nodes with the largest sum. Afterward, the second phase of *BARGS* includes r stages, and each stage allocates the computational resources with $O(m)$ time and generates $O(\frac{T}{r})$ new partial solutions with at most k_{max} nodes for all start nodes. Therefore, the time complexity of the second phase is $O\left(r(m + \frac{T}{r} k_{max})\right) = O(k_{max} T)$, and *BARGS* therefore needs $O(E + m \log n + k_{max} T)$.

4 Experimental Results

We implement *BARGS* in Facebook and invite 50 people from various communities, e.g., schools, government, technology companies, and businesses to join our user study. We compare the solution quality and running time of manual

coordination and *BARGS* for answering PSGA problems, to evaluate the need of an automatic group recommendation service. Each user is asked to plan 5 social activities with the social graphs extracted from their social networks in Facebook. The interest scores follow the power-law distribution with the exponent as 2.5 according to the recent analysis [4] on real datasets. The social tightness score between two friends is derived according to the number of common friends, which represents the proximity interaction [2], and the probability of negative weights [10]. Then, the weighted coefficient λ on social tightness scores and interest scores and the weighted coefficient β on group preference and activity cost in Footnote 4 are set as the average value specified by the 50 people, i.e., $\lambda = 0.527$ and $\beta = 0.514$. Most importantly, after the scores are returned by the above renowned models, each user is allowed to fine-tune the two scores by themselves. In addition to the user study, two real datasets are evaluated in the experiment. The first dataset is crawled from Facebook with $90,269$ users in the New Orleans network[7]. The second dataset is crawled from DBLP dataset with $511,163$ nodes and $1,871,070$ edges.

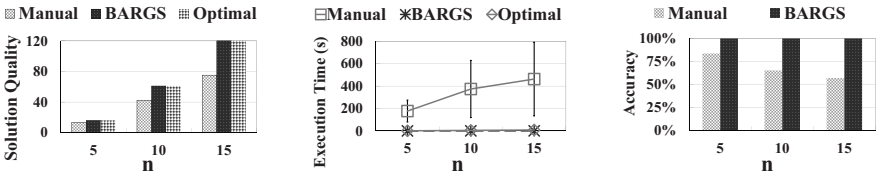
In this paper, the activity cost is modelled by a piecewise linear function, which can approximate any non-decreasing functions. We set the activity cost according to the auditorium cost and other related cost in Duke Energy Center[8].

$$C(k) = \begin{cases} 400 - k & \text{if } 0 \leq k \leq 100. \\ 850 - k & \text{if } 100 < k \leq 600. \\ 2200 - k & \text{if } 600 < k \leq 1750. \end{cases}$$

We compare deterministic greedy (*DGreedy*), randomized greedy (*RGreedy*), and *BARGS* in an HP DL580 server with four Intel E7-4870 2.4 GHz CPUs and 128 GB RAM. *RGreedy* first chooses the same m start nodes as *BARGS*. At each iteration, *RGreedy* calculates the preference increment of adding a neighboring node v_j to the intermediate solution V_S obtained so far for each neighboring node, and sums them up as the total preference increment. Afterward, *RGreedy* sets the node selection probability of each neighbor as the ratio of the corresponding preference increment to the total preference increment, similar to the concept in the greedy algorithm. Notice that the computation budgets represent the number of generated solutions. With more computation budgets, *RGreedy* generates more solutions of group size k_{max}, examines the group utility by subtracting the activity cost from group size 1 to k_{max}, and selects the group with maximum group utility. It is worth noting that *RGreedy* is computationally intensive and not scalable to support a large group size because it is necessary to sum up the interest scores and social tightness scores during the selection of a node neighboring to each partial solution. Therefore, we can only present the results of *RGreedy* with small group sizes. Due to the space constraint, detailed experimental results of the DBLP dataset are presented in [15].

[7] http://socialnetworks.mpi-sws.org/data-wosn2009.html

[8] http://www.dukeenergycenterraleigh.com/uploads/venues/rental/5-rateschedule.pdf



(a) (b) (c)

Fig. 1. Results of user study

The default m in the experiment is set as n/k_{max} since n/k_{max} groups can be acquired from a network with n nodes if each group has k_{max} participants. The default cross-entropy parameters ρ and α are set as 0.3 and 0.99 as recommended by the cross-entropy method [13]. Since *BARGS* natively supports parallelization, we also implemented them with OpenMP for parallelization, to demonstrate the gain in parallelization with more CPU cores.

4.1 User Study

Figures 1(a)-(c) compare manual coordination and *BARGS* in the user study. In addition, the optimal solution is also derived with the enumeration method since the network size is very small. Figures 1(a) and (b) present the solution quality and execution time with different network sizes. The result indicates that the solutions obtained by *BARGS* are identical to the optimal solutions, but users are not able to acquire the optimal solutions even when $n = 5$. As n increases, the solution quality of manual coordination degrades rapidly. We also compare the accuracy of selecting the optimal group size in Figure 1(c). As n increases, it becomes more difficult for a user to correctly identify the optimal size, while *BARGS* can always select the optimal one. Therefore, it is desirable to deploy *BARGS* as an automatic group recommendation service, especially to address the need of a large group in a massive social network nowadays.

4.2 Performance Comparison and Sensitivity Analysis

Figure 2(a) compares the execution time of *DGreedy*, *RGreedy*, and *BARGS* by sampling different numbers of nodes from Facebook data. *DGreedy* is always the fastest one since it is a deterministic algorithm and generates only one final solution, whereas *RGreedy* requires more than 10^5 seconds. The results of *RGreedy* do not return in 2 days as n increases to 10000. To evaluate the performance of *BARGS* with multi-threaded processing, Figure 2(b) shows that we can accelerate the processing speed to 7.2 times with 8 threads. The acceleration ratio is slightly lower than 8 because OpenMP forbids different threads to write at the same memory position at the same time. Therefore, it is expected that *BARGS* with parallelization is promising to be deployed as a value-added *cloud service*.

In addition to the running time, Figure 2(c) compares the solution quality of different approaches. The results indicate that *BARGS* outperforms *DGreedy*

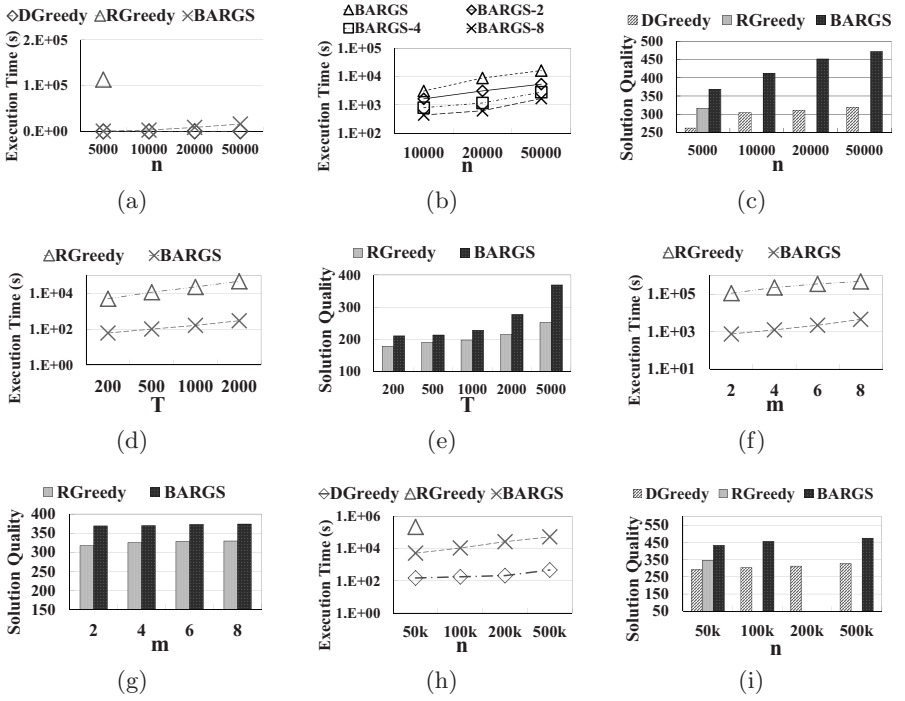


Fig. 2. Experimental results on Facebook and DBLP datasets

and *RGreedy*, especially under a large n. The group utility of *BARGS* is 45% better than the one from *DGreedy* when $n = 50000$. On the other hand, *RGreedy* outperforms *DGreedy* since it has a chance to jump out of the local optimal solution.

Figures 2(d) and (e) compare the execution time and solution quality of two randomized approaches under different total computational budgets, i.e., T. As T increases, the solution quality of *BARGS* increases faster than that of *RGreedy* because it can optimally allocate the computation resources. Even though the solution quality of *RGreedy* is closer to *BARGS* in some cases, *BARGS* is much faster than *RGreedy* by an order of 10^{-2}.

Figures 2(f) and (g) present the execution time and solution quality of *RGreedy* and *BARGS* with different numbers of start nodes, i.e., m. The results show that the solution quality in Figure 2(g) is almost the same as m increases, demonstrating that it is sufficient for m to be set as a value smaller than $\frac{n}{k_{max}}$ as recommended by OCBA [3]. The running time of *BARGS* for $m = 2$ is only 60% of the running time for $m = 4$ as shown in Figure 2(f), while the solution quality remains almost the same.

BARGS is also evaluated on the DBLP dataset. Figures 2(h) and (i) show that *BARGS* outperforms *DGreedy* by 50% and *RGreedy* by 26% in solution quality when $n = 500000$. *BARGS* is still faster than *RGreedy* by an order of 10^{-2}.

However, *RGreedy* runs faster on the DBLP dataset than on the Facebook dataset, because the DBLP dataset is a sparser graph with an average node degree of 3.66. Therefore, the number of candidate nodes to be chosen during the expansion of the partial solution in the DBLP dataset increases much more slowly than in the Facebook dataset with an average node degree of 26.1. Nevertheless, *RGreedy* is still not able to generate a solution for a large network size n due to its unacceptable efficiency.

5 Conclusion

To the best of our knowledge, there is no real system or existing work in the literature that addresses the issues of scale-adaptive group optimization for social activity planning based on topic interest, social tightness, and activity cost. To fill this research gap and satisfy an important practical need, this paper formulated a new optimization problem called PSGA to derive a set of attendees and maximize the group utility. We proved that PSGA is NP-hard and devised a simple but effective randomized algorithms, namely *BARGS*, with a guaranteed performance bound. The user study demonstrated that the social groups obtained through the proposed algorithm implemented in Facebook significantly outperforms the manually configured solutions by users. This research result thus holds much promise to be profitably adopted in social networking websites as a value-added service.

Acknowledgments. This work is supported in part by NSF through grants CNS-1115234, and OISE-1129076.

References

1. Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. IEEE Transactions on Knowledge and Data Engineering **20**, 172–188 (2008)
2. Chaoji, V., Ranu, S., Rastogi, R., Bhatt, R.: Recommendations to boost content spread in social networks. In: WWW, pp. 529–538 (2012)
3. Chen, C.H., Yucesan, E., Dai, L., Chen, H.C.: Efficient computation of optimal budget allocation for discrete event simulation experiment. IIE Transactions **42**(1), 60–70 (2010)
4. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM **51**(4), 661–703 (2009)
5. Deutsch, M., Gerard, H.B.: A study of normative and informational social influences upon individual judgment. JASP **51**(3), 291–301 (1955)
6. Feng, K., Cong, G., Bhowmick, S.S., Ma, S.: In search of influential event organizers in online social networks. In: Proc. SIGMOD (2014)
7. Gleich, D.F., Seshadhri, C.: Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In: KDD, pp. 597–605 (2012)
8. Kaplan, M.F., Miller, C.E.: Group decision making and normative versus informational influence: Effects of type of issue and assigned decision rule. JPSP **53**(2), 306–313 (1987)

9. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: CIKM, pp. 985–994 (2011)
10. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: CHI (2010)
11. McClosky, B., Hicks, I.V.: Combinatorial algorithms for max k-plex. Journal of Combinatorial Optimization (2012)
12. Mitzenmacher, M., Upfal, E.: Probability and computing: Randomized algorithms and probabilistic analysis. Cambridge University Press (2005)
13. Rubinstein, R.Y.: Combinatorial optimization, cross-entropy, ants and rare events. In: Uryasev, S., Pardalos, P.M. (eds.) Stochastic Optimization: Algorithms and Applications, pp. 304–358. Kluwer Academic (2001)
14. Saríyüce, A.E., Gedik, B., Jacques-Silva, G., Wu, K.-L., Çatalyüreks, U.V.: Streaming algorithms for k-core decomposition. VLDB **6**(5), 433–444 (2013)
15. Shuai, H.-H., Yang, D.-N., Yu, P.S., Chen, M.-S.: Scale-adaptive group optimization for social activity planning. In: CoRR (1502.06819) (2015)
16. Xiang, J., Guo, C., Aboulnaga, A.: Scalable maximum clique computation using mapreduce. In: ICDE, pp. 74–85 (2013)
17. Yang, D.N., Shen, C.Y., Lee, W.C., Chen, M.S.: On socio-spatial group query for location-based social networks. In: KDD, pp. 949–957 (2012)
18. Ye, M., Liu, X., Lee, W.C.: Exploring social influence for recommendation - a probabilistic generative model approach. In: SIGIR, pp. 671–680 (2012)